International Journal of Computer Vision and Signal Processing, 6(1), 20-32(2016)

View invariant vehicle type recognition and counting system using multiple features

Martins E. Irhebhude*

Department of Computer Science, Nigerian Defence Academy, Kaduna, Nigeria

Amin Nawahda

Environmental Research Centre, Sohar University, Oman

Eran A. Edirisinghe

Department of Computer Science, Loughborough University, UK

Abstract

This paper presents an automatic vehicle type recognition, classification and counting system that is view invariant and can be used in real-time road transportation and environmental pollution management systems. A Support Vector Machine (SVM) classifier is used to classify vehicles into four categories namely; cars, jeeps, buses and trucks, respectively, based on measurable image features. Image analysis is performed on a set of features that consists of Region, Histogram Oriented Gradient (HOG) and Local Binary Pattern (LBP) histogram features. A feature combination approach is proposed for recognition; Region, LBP and HOG (RLH). Correlation based Feature Selection (CFS) is used to select the most discriminative features from the above feature set thereby improving recognition accuracy and reducing the time required for classification. Various success rates are reported from the experiments conducted on two separate datasets, with average accuracy reaching 95% on the combined datasets. Proposed feature combination techniques compared with Region, HOG, LBP, with results showing higher recognition accuracy achievable when using the proposed feature combinations. Separate frontal/rear and angular view datasets have been used in the experiments to demonstrate the adaptability of the proposed algorithm to variations of the view angle. Initially three practical application scenarios that will benefit from the proposed technology are presented namely; access control, toll collection and the estimation of air pollution levels caused by vehicular traffic, justifying the practical importance and relevance of the research presented.

Keywords: Region descriptors, Local Binary Pattern, Histogram Oriented Gradients, Machine Learning, Intelligent Transport System, Vehicle Type Recognition

IJCVSP

ISSN: 2186-1390 (Online)
http://cennser.org/IJCVSP

Article History:
Received: 2 Feb. 2016
Revised: 14 April 2016
Accepted: 10 Nov. 2016
Published Online: 11 Nov. 2016

1. INTRODUCTION

In the recent past, concerns directly associated with vehicle related crime have risen internationally. As described by the Interpol report of [1] vehicle crime is:

> *...a highly organized criminal activity affecting all regions of the whole world and with clear*

*Corresponding author
Email addresses: M.E.Irhebhude@lboro.ac.uk, hyelmart4lyf@gmail.com (Martins E. Irhebhude), ANawahda@soharuni.edu.com (Amin Nawahda), E.A.Edirisinghe@lboro.ac.uk (Eran A. Edirisinghe)

links to organized crime and terrorism. Vehicles are not only stolen for their own sake, but are also trafficked to finance other crimes. They can also be used as bomb carriers or in the perpetration of other crimes.

To this end, there are many situations where access to vehicles needs to be monitored automatically to manage and control their movement to and from secure sites, motorways and across international borders. Although number plate recognition provides a level of security based on the licence information gathered, in an era where vehicle cloning prevails, any additional vehicle identification data can help to improve the robustness against such unlawful activities. The identification of vehicle type and keeping a count of each type passing certain known locations will help this process.

Further the increase of the cost of building and maintaining motorways have forced many governments to consider privatising motorways resulting in a need for toll collection from their users. The number of toll roads present is growing fast internationally and so is the crime rate to avert the payment of the correct toll. Toll is normally charged based of vehicle type and the varied tariffs used means that when a human observer is not present the systems can be fooled by vehicle that is charged a higher rate being driven through a gate meant to be for a type that is charged less. The automatic identification of the vehicle type can help take preventive measures to stop this crime.

The exponential increase of road traffic over the years has caused serious concerns about the level of pollution caused by vehicular traffic. Especially the production of ozone is considered as a by-product of nitrogen dioxide caused directly due to internal combustion engines, when exposed to direct sunlight. The larger the power of a vehicle engine, the larger would be the impact it will have on the creation of the secondary pollutants such as, ozone. The counting of various types of vehicles that uses a motorway on an hourly or daily basis will help in estimating the emitted and formed air pollutants from vehicles [2]. Simply detecting vehicles and tracking them will allow the monitoring of total road usage and the estimation of their speed that also has an impact on approximating the pollution levels [2].

The above needs solicits the importance of the design, development, implementation and the installation of a computer vision based automated vehicle counting and type recognition system, which is the key focus of the research presented in this paper.

Existing literature in vehicle detection, counting and type recognition proposes a number of different approaches. The authors of [3] showed that even in a congested road traffic condition an AND-OR graph (AOG) using bottom-up inference can be used to represented and detect vehicle objects based on both frontal and rear views. In a similar situation, [4] proposed the use of strong shadows as a feature to detect the presence of vehicles in a congested environment. In [5], vehicles were partitioned into three parts; road, head and body, using a tripwire technique. Subsequently haar wavelet features extracted from each part and principal component analysis (PCA) is performed on these features calculated to form 3 category PCA-subspaces. Further multiple discriminate analysis (MDA) is performed on each PCA-subspace to extract features which are subsequently trained to identify vehicles using the hidden markov model-expectation maximisation HMMEM algorithm. In another experiment, a camera calibration tool was used on detected and tracked vehicle objects so as to extract object parameters, which were then used for the classification of the vehicle into the class of cars or non-cars [6]. In [7] vehicle objects were detected and counted using a frame difference technique with morphological operators, dilation and erosion. In [8], using maximum likelihood Bayes decision rule classifier on normalised local features (roof, two tail-lights and headlights of rear and front view) vehicle or non-vehicle objects are detected. Further to handling the unevenness of the road surface, the author simulated images with PCA applied on each sub-region to reduce feature sets, computation time and to speed-up processing cycle. In another classification task by [9], segmentation through image differencing was used to obtain foreground object upon which sobel edge is computed; furthermore, size feature is extracted from two level dilations with filling morphological image sets and used for classification into small, medium and large. In [10], an alternative to expensive electronic toll collection (ETC) full-scale multi-lane free flow traffic system was proposed; the technique used scale-invariant feature transform (SIFT), Canny edge detector, k-means clustering with Euclidean matching distance metric for inter and intra class vehicle classification. A technique for rear view vehicle classification, [11] proposed the use of a hybrid dynamic Bayesian Network (HDBN) to classify vehicles. Tail light and vehicle dimensions with respect to the dimensions of the license plate were the feature sets used for classification. Width distance from license plate and the angle between the tail light and the license plates formed the eleven features used for classification. The experiment was performed in two phases; known vs unknown classes and four known classes using HDBN. HDBN was compared with three other classifiers. The performance evaluation result using a ROC curve shows HDBN as the best classifier for rear view vehicle classification. In [12], a technique for traffic estimation and vehicle classification using region features with a neural network (NN) classifier was proposed.

In observing the techniques proposed in literature summarised above, it can be concluded that vehicles are recognised and classified at different angles under different conditions using different feature sets, classification techniques and hence algorithms. In other words a change of camera angle requires a change of features that needs to be extracted for classification. The classification technique that performs best will also change. Further, most techniques

have been tested either on rear or front views only. In practice once a camera is installed in an outdoor environment with the hope of capturing video footage for vehicle type recognition, it is likely that due to wind or neglect in installation, the camera could turn in due course. If the vehicle type recognition system was dependent significantly on the angle of view, the system would thus fail to operate accurately. Further at the point of installation practical problems may be such that the camera position and orientation will have to be changed as compared to the fixed angular view that it has originally being designed for. This will either requires the system to be re-redesigned using different feature sets, classifiers and algorithms or the system having to go through a camera calibration processes which is typically non-trivial and time consuming. It would be ideal if at the new orientation the captured content could still be used for classification.

Given the above observations we propose a novel algorithm for vehicle type recognition and subsequent counting, which is independent of the camera view angle. We adopt a strategy that uses multiple features that are scale and rotation invariant, leading to the accurate classification of vehicles independent of the camera angle.

For clarity of presentation this paper is divided into five sections. Apart from this section which introduces the reader to the problem domain and identifies the research gap based on existing work presented in the literature, the remaining sections are structured as follows: the research background is presented in section 2 and the proposed vehicle type recognition algorithm is presented in section 3. Section 4 is focused on experimental results and a performance analysis while section 5 shows comparison with state of art techniques with concluding remarks and further work provided in section 6.

2. BACKGROUND OF STUDY

The novel algorithm to be presented in next section and illustrated by Figure 1 is based on a number of established theories and mathematical concepts. To this effect this section provides the theoretical foundations to Gaussian Mixture Models (GMM), Canny Edge Detection (CED), Local Binary Pattern (LBP), Histogram of Oriented Gradients (HOG), Correlation based Feature Selection (CFS) and Support Vector Machines (SVM). The foundations of the proposed approaches are built upon these concepts, theories and mathematical definitions.

2.1. Gaussian Mixture Model (GMM)

According to [13], a GMM is a parametric probability density function that is represented as a weighted sum of Gaussian distributions. The GMM technique uses a method to model each background pixel by a mixture of k Gaussian distributions [14]. The weight of the mixture represents the time proportion for which the pixel values stay unchanged in a scene. Probable background colours stay longer and are more static than the foreground colours.

In [13], the recent history of each pixel, $X_1, ..., X_t$, is modelled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is defined as:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \sum_{i,t}) \tag{1}$$

where K is the number of distributions, $\omega_{i,t}$ is an estimate of the weight (what portion of the data is accounted for by this Gaussian) of the i^{th} Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i^{th} Gaussian in the mixture at time t, $\sum_{i,t}$ is the covariance matrix of the i^{th} Gaussian in the mixture at time t, and η is a Gaussian probability density function of the form:

$$\eta(X_t, \mu, \sum) = \frac{1}{(2\pi)^{\frac{n}{2}} |\sum|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu_t)^T \sum^{-1} (X_t - \mu_t)} \tag{2}$$

The covariance matrix is of the form:

$$\sum_{k,t} = \sigma_k^2 I \tag{3}$$

2.2. Canny Edge Detection

According to [15] an edge detector is an operator that is sensitive to grey level change in an image. Detecting these changes in intensity can be accomplished using first or second-order derivatives [16]. Finding edge strength and direction at location (x, y) of an image, I, is accomplished using the gradient, denoted by ∇I; defined by the vector [16]:

$$\nabla I \equiv grad(I) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \end{bmatrix} \tag{4}$$

Equation 4 has an important geometrical property that it points in the direction of the greatest rate of change of I at location (x, y).

The direction measured with respect to the $x - axis$ and the value of the rate of change in the direction of the gradient vector is denoted as [16]:

$$M(x, y) = mag(\nabla I) = \sqrt{g_x^2 + g_y^2} \tag{5}$$

and

$$\alpha(x, y) = tan^{-1} \left[\frac{g_y}{g_x} \right] \tag{6}$$

Canny defined a set of goals for an edge detector and described an optimal method for achieving them [15]. The goals are,

- Error rate: a detector should respond to edges only and not miss any.

- Localisation: the distance between pixels found and the actual edge should be as small as possible.
- Response: should not identify multiple edges where only one edge is present.

Canny assumed a step edge subject to white Gaussian noise [15]. The edge detector was assumed to be a convolution filter f, which would smooth the noise and locate the edge.

In order to capture the contour in the vehicle types, we applied CED on the vehicle object so as to be able to subsequently encode shape information based on edges.

2.3. Local Binary Pattern (LBP)

The Local Binary Pattern (LBP) operator labels the pixels of an image with decimal numbers that encode the local structure around each pixel of an image [17]. Each pixel (i.e. $g^1, g^2, \cdots, g^8$) is compared with its eight neighbours (see equation 7) by subtracting the center pixel value; the results; if negative, are encoded as 0, and the otherwise 1 (see equation 8). For each given pixel, a binary number is obtained by concatenating all these binary values (referred to as LBPs, see equation 9) in a clockwise direction, which starts from the one of its top-left neighbour. The corresponding decimal value of the generated binary number is then used for labeling the given pixel.

LBP can be described as follows:

Pixel neighbourhood:

$$\begin{pmatrix} g_8 & g_1 & g_2 \\ g_7 & g_c & g_3 \\ g_6 & g_5 & g_4 \end{pmatrix} \tag{7}$$

thresholding:

$$\begin{pmatrix} s(g_8-g_c) & s(g_1-g_c) & s(g_2-g_c) \\ s(g_7-g_c) & & s(g_3-g_c) \\ s(g_6-g_c) & s(g_5-g_c) & s(g_4-g_c) \end{pmatrix} s(x) = \begin{cases} 1, x \geq 0 \\ 0, x < 0 \end{cases} \tag{8}$$

LBP for pixel:

$$LBP = \sum_{p=0}^{P-1} s(g_p - g_c) 2^P$$

Example

$$\begin{pmatrix} 56 & 58 & 95 \\ 20 & 80 & 98 \\ 22 & 79 & 80 \end{pmatrix}$$

$$\begin{pmatrix} s(56-80) & s(58-80) & s(95-80) \\ s(20-80) & & s(98-80) \\ s(22-80) & s(79-80) & s(80-80) \end{pmatrix} \Rightarrow \begin{pmatrix} 0 & 0 & 1 \\ 0 & & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 0\times 2^7 & 0\times 2^0 & 1\times 2^1 \\ 0\times 2^6 & & 1\times 2^2 \\ 0\times 2^5 & 0\times 2^4 & 1\times 2^3 \end{pmatrix} \Rightarrow 00001110_2 \tag{9}$$

$$00001110_2$$
$$\Rightarrow 14$$

Other variants are; circular LBP, rotation invariant LBP, uniform LBP, multi-scale LBP, and multi-dimensional LBP.

2.4. Histogram of Oriented Gradient (HOG)

According to [18], HOG is described as a concept that the local appearance and shape of an object can be characterized well by the distribution of local intensity gradients or edge direction, without knowledge of edge positions. It is usually implemented by dividing an image window into small regions named cells and accumulating each local cell's $1-D$ histogram of gradient directions or edge orientations over the pixels. The combined entries form this representation that is contrast normalised to ensure invariance to illumination. This normalisation is extended to all cells in the block to form the HOG descriptor. Dalal and Triggs [18] explored different methods for block normalization. Let v be the non-normalized vector containing all histograms in a given block, $\|v\|_k$ be its $k-norm$ for $k = 1, 2$ and e be a small constant. Then the normalization factor can be one of the following:

$$L2-norm : f = \frac{v}{\sqrt{\|v\|_2^2 + e^2}} \tag{10}$$

$L2-hys$: $L2-norm$ followed by clipping (limiting the maximum values of v to 0.2) and renormalizing,

$$L1-norm : f = \frac{v}{\|v\|_1 + e} \tag{11}$$

$$L1-sqrt : f = \sqrt{\frac{v}{\|v\|_1 + e}} \tag{12}$$

In their experiments, Dalal and Triggs found that the $L2-Hys$, $L2-norm$, and $L1-sqrt$ schemes provide similar performance, while the $L1-norm$ provides a slightly less reliable performance; however, all four methods showed significant improvement over the non-normalized data [18].

2.5. Correlation-based Feature Selection

According to [19] as reported by [20], CFS is a filtering algorithm that evaluates subset features based on the individual feature predicting power of a class label. In [20] CFS is reported as:

$$M_s = \frac{k.\overline{r_{cf}}}{\sqrt{k + k.(k-1).\overline{r_{ff}}}} \tag{13}$$

where k is the number of features selected in current subset, r_{cf} is the mean feature-class correlation for each element of current subset, r_{ff} is the mean feature-feature correlation for each pairwise of element. It begins with empty set and one at a time add features that holds best value. Best first search method is applied to get merit value.

2.6. Support Vector Machine

According to [21], SVM is a technique used to train classifiers, regressors and probability densities that is well-founded in statistical learning theory. SVM can be used for binary and multi-classification tasks.

2.6.1. Binary classification

SVM perform pattern recognition for two-class problems by determining the separating hyperplane with maximum distance to the closest points of the training set. In this approach, optimal classification of a separable two-class problem is achieved by maximising the width of the margin between the two classes [22]. The margin is the distance between the discrimination hyper-surface in n-dimensional feature space and the closest training patterns called support vectors. If the data is not linearly separable in the input space, a non-linear transformation $\Phi(.)$ can be applied, which maps the data points $x \in \mathbb{R}$ into a high dimensional space H, which is called a feature space. The data is then separated as described above. The original support vector machine classifier was designed for linear separation of two classes; however, to solve the problem of separating more than two classes, the multi-class support vector machine was developed.

2.6.2. Multi-class classification

SVM was designed to solve binary classification problems. In real world classification problems however, we can have more than two classes. In the attempt to solve $qclass$ problems with SVMs; training q SVMs was involved, each of which separates a single class from all remaining classes, or training q^2 machines, each of which separates a pair of classes. Multi-class classification allows non-linearly separable classes by combining multiple $2-class$ classifiers. $N-class$ classification is accomplished by combining N $2-class$ classifiers, each discriminating between a specific class and the rest of the training set [22]. During the classification stage, a pattern is assigned to the class with the largest positive distance between the classified pattern and the individual separating hyperplane for the N binary classifiers. One of the two classes in such multi-class sets of binary classification problems will contain a substantially smaller number of patterns than the other class [22].

SVM implementation of [23] was used in the experiments supporting the research presented in this thesis, using CS model type, Gaussian RBF kernel and mean normalised parameters.

SVM classifier was chosen because of its popularity and speed of processing.

3. RESEARCH METHODOLOGY

This section introduces the reader to the proposed methodology, presenting in detail the functionality of each module/stage of the proposed vehicle type recognition system under three main topics: vehicular object segmentation; feature extraction; and vehicular object classification. Figure 1 illustrates a block diagram of the proposed system.

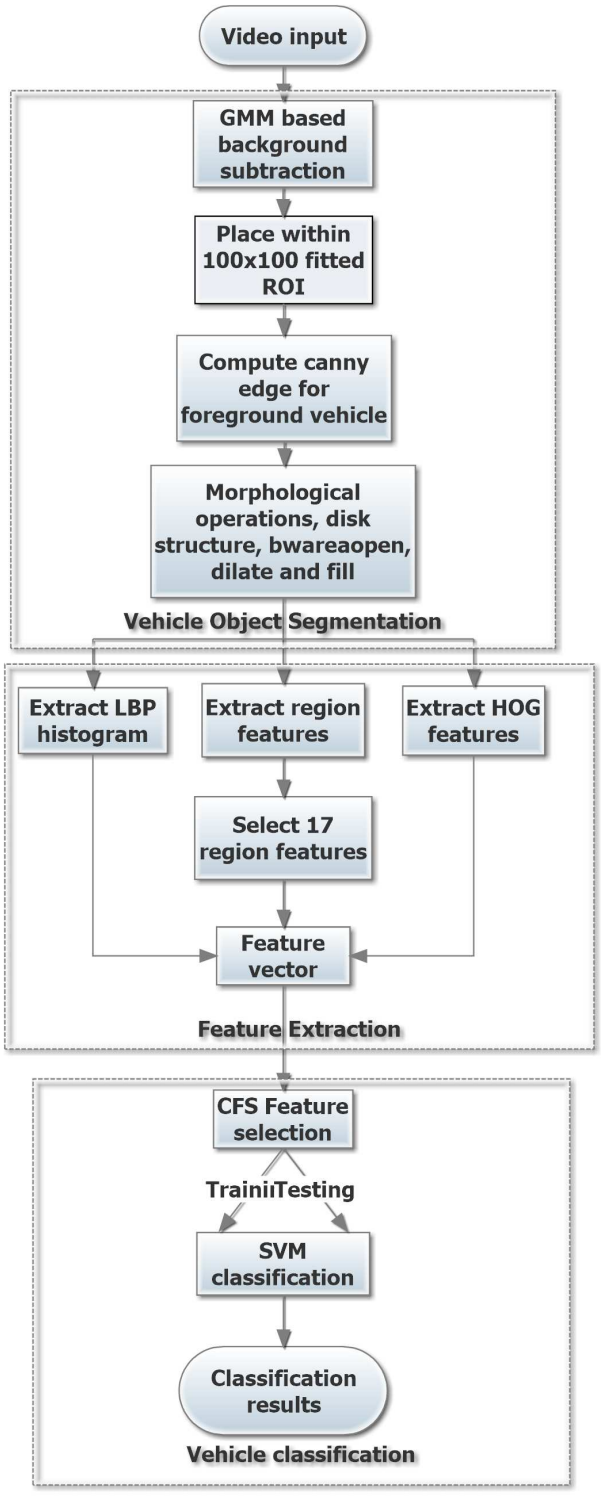

Figure 1: Proposed methodology for vehicle classification

The analysis of the performance of the proposed system for vehicle type recognition was conducted on datasets gathered from two low medium resolution cameras that were installed on the roadside of the Sohar Highway, Oman.

They were of pixel resolution 320×240, and the frame rate was 25 FPS. The data used in the experimental analysis consisted of 10 hours video footage, captured during daytime.

3.1. Vehicular object segmentation

The video frames were first segmented using a GMM (section 2.1) based foreground/background subtraction algorithm [14, 13] that detects moving objects. Due to the close-up view settings used in capturing the video footage from a motorway environment, it can be assumed that all foreground objects picked up by the above algorithm are moving vehicles only. The segmented vehicular object regions need further processing to ensure that the segmented regions more appropriately represent the true shape of a vehicle. For this purpose a Canny Edge detector was first used to estimate the edges of the segmented object and the segmented region was subsequently refined using several morphological operators [24] that included, disk structure, bwareaopen, dilation and filling. The contribution of each of the operators in improving the segmented vehicular object shapes is demonstrated by the experimental results presented in figure 2.

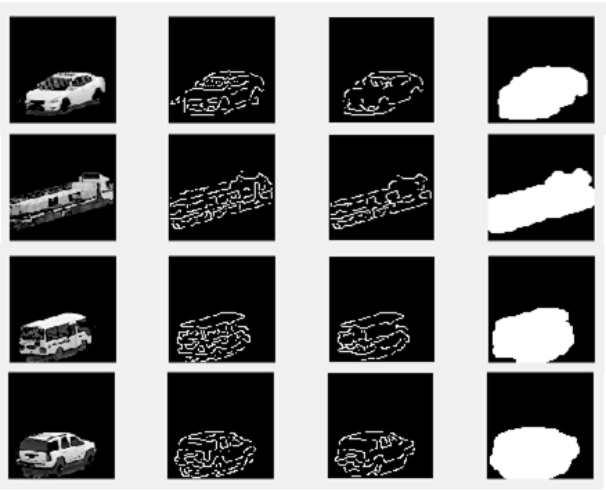

Figure 2: Diagram shows original vehicle, after edge detection, after removing extra edges and after dilate and fill operations respectively

After the extraction of the foreground vehicular objects, they are placed within tightest fitting square shaped Regions of Interest (RoIs). These are subsequently resized to a normalised size of 100×100 pixels that are the regions used by subsequent stages for further processing.

3.2. Feature extraction

Feature extraction is performed on square shaped windows (normalised to 100×100 pixel areas) surrounding the segmented foreground objects, with the background pixels within the square area set to zero. Firstly for the purpose of training, we manually extracted vehicle image samples normalised to a size of 100×100 pixels from the recorded video footage frames. Figures 3 and 4 illustrate some examples of segmented foreground objects.

Figure 3: Training samples - some segmented vehicles from front/rear view dataset

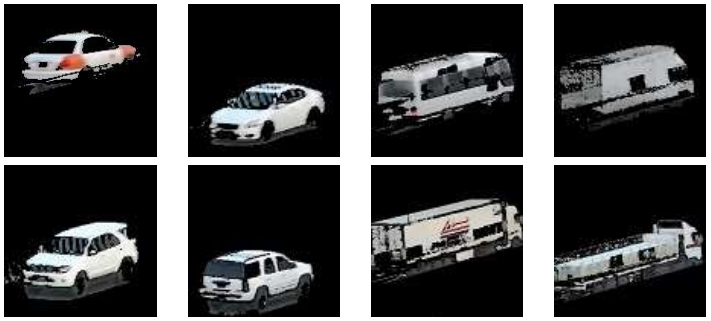

Figure 4: Training samples - some segmented vehicles from angular view dataset

For testing purposes the regions of interest are automatically segmented following the process described in section 2.1 Note that once the segmented foreground region is extracted it is first enclosed within a tightest fitting square area that is subsequently normalised to a size 100×100 pixels. The pixels within the square area but outside the object's region-of-interest is set to zero. The features are calculated on the above mentioned 100×100 square regions. The following sections describe the feature extraction process.

We propose the use of 17 simple scalar Region descriptors as features, alongside Histogram Oriented Gradient (HOG) and Local Binary Pattern (LBP) histogram features. They can be detailed as follows.

3.2.1. Region Discriptors/Features

: We propose the initial use of 17 Region Features, which can be defined as follows:

1. Area[25]: The total number of pixels that are included in the region-of-interest within the square area.
2. Centroid [25]: Horizontal and vertical coordinates of center of mass are computed as the two feature that represent the centroid.
3. Bounding Box [26]: The smallest rectangle containing the region-of-interest. Bounding box feature is of the form $[x,\ y,\ width]$; where $x,\ y$ specifies the upper-left corner of the bounding box, and width is in the form $[x_{width}\ y_{width}...]$ and specifies the length of the bounding box along each dimension.

4. Eccentricity [26]: The Eccentricity characteristic is the ratio of the length of the maximum chord A to the maximum chord B, which is perpendicular to of the region-of-interest enclosed within the rectangle.
5. Major Axis Length [26]: The length (in pixels) of the major axis of the ellipse that has the same second moments as the region-of-interest.
6. Minor Axis Length [26]: the length (in pixels) of the minor axis of the ellipse that has the same second moments as the region-of-interest.
7. Orientation [26]: The angle (in degrees) between the x-axis and the major axis of the ellipse that has the same second-moments as the region-of-interest.
8. Filled Area [26]: The number of pixels in Filled Image. where filled image is a binary image (logical) of the same size as the bounding box of the region-of-interest.
9. Convex Area [26]: The number of pixels within the convex hull of the region-of-interest, with all pixels within the hull filled in.
10. EquivDiameter [26]: the diameter of a circle having the same area as the region-of-interest.
11. Solidity [26]: The proportion of the pixels in the convex hull that are also within the region-of-interest. Computed as Area / ConvexArea.
12. Extent [26]: The proportion of the pixels in the bounding box that are also in the region-of-interest. Computed as the Area divided by area of the bounding box.
13. Perimeter [26]: The perimeter is the length of the boundary of the object region-of-interest, in pixels.

Note that since horizontal and vertical coordinates of the centroid are computed as two separate centroid features and bounding box features includes four component features, namely, the x and y co-ordinates of the top left hand corner and the width and height of the bounding box there are altogether a total of 17 region features that will be considered.

3.2.2. HOG Feature

The HOG features were extracted as defined in section 2.4. HOG feature of length 144 was computed thus:

$$\begin{aligned} \vec{x} &= (I./\vec{z} - \vec{y})./(\vec{y} - \vec{ol}) + 1 \\ length &= \prod[\vec{x}, \vec{y}, k] \end{aligned} \tag{14}$$

where $\vec{y}$ is a two element vector [2 2] (block size), k is the number of bins; 9 , $\vec{z}$ is a two element vector [32 32] (cell size), I is 100×100 (size of the image), and $\vec{ol}$ is [1 1] ($\vec{y}/2$ - block overlap).

3.2.3. LBP Feature

The LBP histogram Feature (section 2.3) was extracted from each image enclosed within the 100×100 rectangular area giving a 256 bin histogram.

3.2.4. Feature Combination

In order to recognise, classify and count vehicle types, we captured appearance and shape information using the proposed feature sets; in doing so, Region Features, HOG Features, and LBP Histogram Features defined above were extracted from the segmented foreground object and were combined to form a feature attributes for the classification of vehicles into four categories namely, cars, buses, jeeps and trucks respectively. The extracted Region (17), HOG (144) and LBP histogram (256) features were combined and used for the experiments.

3.3. Feature selection

To reduce the feature space and speed-up the processing cycle, we used the CFS [19] approach (see Section 2.5) as feature selector. CFS algorithm helps to rank feature subsets according to the correlation based on the heuristic "merit" as reported by [20]. This reduced the original feature attributes obtained from the segmented foreground vehicle objects to minimal features attributes. In section 4 we showed that with feature selection, substantial accuracy improvement for vehicle classification using both types of views was achieved.

4. EXPERIMENTAL ANALYSIS

A number of experiments were conducted to evaluate the performance of the proposed algorithm in vehicle type recognition. The experiments were conducted on video footage captured by a general purpose, non-calibrated, CCTV camera installed on the side of Sohar Highway, Oman, in the city of Sohar. As the robustness of the algorithm to the vehicle's angle of approach to the camera axis and real-time performance capability are two important design criteria, further experiments were conducted to evaluate in detail the accuracy and speed of the proposed algorithm.

Two video datasets were collected for training and testing, by installing the camera appropriately to capture front/rear (F/R) views of the vehicles and side/angular views. The first dataset was collected during a short duration (15 minutes) and captured the views of the vehicles in line with the motorway lanes. This was achieved by filming from an overhead bridge with the camera installed rigidly on a tripod. The second dataset was captured over a 10 hour period of daytime and recorded footage at approximately a 45^0 angle from the direction of the movement of vehicles. It is this angle that we consider a more practical direction of view for a camera installed in the roadside. The experimental results for the two datasets are presented, combined and separated to enable subsequent, direct comparison. The idea is to prove that the proposed algorithm can produce accurate results regardless of the angle of operation as long as training has been done on sample images that have been recorded at a similar angle.

In general, the set of input-output sample pairs that are used for the training of the classifier can be represented as,

$$(x_1, y_1), (x_2, y_2), ..., (x_N, y_N) \quad (15)$$

where the input x_i denotes the feature vector extracted from image I and the output y_i is a class label. Since we are categorising into vehicle types, the class label y_i encodes the four vehicle types, namely, cars, buses, jeeps and trucks; while the extracted feature x_i encodes one of the combinations of the feature sets described above, i.e. Region, LBP and HOG features; 1). Region; 2). LBP; 3). HOG; 4). Region and LBP (RL) ; 5). Region and HOG (RH); 6). Region, LBP and HOG (RLH); 7). LBP and HOG (LH) respectively.

Note that all of the above seven feature set combinations were tested to determine which combinations results in the best accuracy. As two datasets were used, namely; F/R view dataset and angular view dataset. The experimental results are presented separately in sections 4.1 and 4.2 respectively, as follows:

4.1. Experiments on the front and rear view dataset

The dataset consisted of approximately 100 different vehicles and was split 50:50 for the purpose of training and testing. The vehicles captured and thus used in experimentation only consisted of two vehicle types, namely, cars and buses (unfortunately due to short duration of test data recording no jeeps and trucks were captured) and hence the classification was of a binary nature, i.e. into these two classes.

We conducted experiments using different feature attributes; 1). Region; 2). LBP; 3). HOG; 4). RL; 5). RH; 6). RLH; 7). LH. Various success rates were recorded. Using region features, we recorded 93% prediction accuracy when using the entire set of feature attributes and the same percentage accuracy when CFS selected 3 discriminating features from the original 17. Using LBP features only, we recorded 79% recognition accuracy using the entire set of feature attributes with significant improvement of recognition accuracy to 90% when CFS selected 8 discriminating features from the original 256. Using HOG features only, we recorded a 97% recognition accuracy using the entire set of feature attributes with accuracy dropping to 94%, when CFS selected 23 discriminating features, from the original set of 144. Using RL features, we recorded 99% recognition accuracy using the entire set of feature attributes with improvement to 100% recognition accuracy when CFS selected 8 discriminating features from the possible total of 273. Using RH features, we recorded 96% recognition accuracy using the entire set of feature attributes, with an improvement to 97% recognition accuracy when CFS selected 10 discriminating features from the total of 161. Using LH features, we recorded 96% recognition accuracy using the entire set of feature attributes, with an improvement to 97% recognition accuracy when CFS selected 24 discriminating features from a total of 400. Finally, using RLH features, we recorded 97% recognition accuracy when using the entire set of feature attributes with same level of accuracy of 97% indicated when CFS selected 16 discriminating features from the original 417.

A summarisation of these results and observations are recorded in the first third of the table 1.

Figure 5 below shows an example of a classified vehicle from the F/R dataset.

Figure 5: Some examples of recognised vehicle from front/rear dataset

4.2. Experiments on angular view dataset

The second dataset obtained at an angle of approximately 45^0 to the direction of vehicular movement consisted of sufficient number of examples of all four types of vehicles that can be used for training purposes. Therefore the classification was carried out into four categories cars, jeeps, trucks and buses respectively. A 50:50 split was used for training and testing.

We conducted experiments using all of the seven different selections of feature attributes; 1). Region; 2). LBP; 3). HOG; 4). RL; 5). RH; 6). RLH; 7). LH. Various success rates were recorded. Using region features only, we recorded 85.7% recognition accuracy using the entire set of feature attributes with an improvement to 86% recognition accuracy when CFS selected 9 discriminating features from the original 17. Using LBP features, we recorded 74% recognition accuracy using the entire set of feature attributes, with a significant improvement to 77% recognition accuracy when CFS selected 20 discriminating features from the original 256. Using HOG features, we recorded 92.7% recognition accuracy using the entire set of feature attributes with accuracy dropping to 89% recognition accuracy when CFS selected 34 discriminating features from the original 144. Using RL features, we recorded 89% recognition accuracy using the entire set of feature attributes with an improvement to 96% recognition accuracy when CFS selected 26 discriminating features from the original 273. Using RH features, we recorded a 95% recognition accuracy using the entire set of feature attributes with the accuracy dropping to 93% when CFS selected 22 discriminating features from the original 161. Using LH features, we recorded a 93% recognition accuracy using the entire set of feature attributes with an improvement to 94.7%

recognition accuracy when CFS selected 47 discriminating features from the original 400. Finally, using RLH features, we recorded a 94% recognition accuracy using the entire set of feature attributes with significant improvement of accuracy to 97% when CFS selected 37 discriminating features from the original set of 417.

A summarisation of these results and observations are recorded in the second third of the table 1.

Figure 6 below shows some examples of classified vehicles from the angular view dataset.

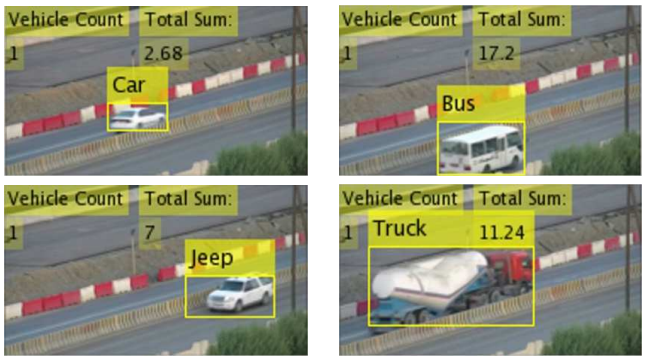

Figure 6: Some examples of recognised vehicle from angular dataset

4.3. Analysis of Results

Table 1 summarises the recognition accuracies achieved when using the two datasets (i.e. F/R and angle), with and without feature selection. It also gives an indication of the number of features in each category, i.e., Region, HOG, LBP and their various feature combinations that remain after feature selection is applied. The table also includes experimental results when the two datasets were combined for both training and testing purposes. [Note: these are included in the bottom third of the table. CV = combined view.].

Table 1: Classification accuracy results with selected features

Features	Whole acc	Selected acc	Selected features	View angle
HOG	97%	94%	23	F/R
Region	93%	93%	3	F/R
LBP	79%	90%	8	F/R
RL	99%	100%	4R,4L	F/R
RH	96%	97%	3R,7H	F/R
LH	96%	97%	6L,18H	F/R
RLH	97%	97%	3R,8H,5L	F/R
HOG	92.7%	89%	34	Angle
Region	85.7%	86%	9	Angle
LBP	74%	77%	20	Angle
RL	89%	96%	9R,17L	Angle
RH	95%	93%	8R,14H	Angle
LH	93%	94.7%	22L,25H	Angle
RLH	94%	97%	7R,14H,16L	Angle
HOG	90%	87.8%	35	CV
Region	75%	74%	7	CV
LBP	74%	80%	23	CV
RL	84%	82.5%	7R,15L	CV
RH	91.5%	83.5%	9R,11H	CV
LH	89.5%	91.8%	20L,25H	CV
RLH	91.5%	90.8%	8R,12H,15L	CV

The overall conclusion when observing the results tabulated in table 1 is that the feature combinations RL, RH, LH and RLH performs best as against using a single set of features all being either Region, LBP or HOG features.

Results tabulated in table 1 shows that the experiments on the first dataset (that consists of vehicles captured from their F/R) indicates higher accuracy figures as compared to experiments with the second dataset (angular view). There are many reasons for this. It is noted that with the F/R dataset the classifications were done only between two classes, namely cars and buses. This was due to the practical reason that during the short duration (15 minutes) in which the video footage of F/R dataset was captured, only a very few samples of trucks and jeeps appeared in the footage. This made it impossible to find sufficient samples to train the classifier. Classifying between two vehicle classes which are relatively distinct (i.e. cars vs buses) as in the experiments, will be more accurate as compared to discriminating between four vehicular classes that have some class pairs, which are harder to discriminate between (e.g. cars vs jeeps and jeeps vs mini buses). This argument is justified when analysing the confusion matrices of tables 2and 3 for the two datasets using the feature set of RLH. Further the angular dataset was significantly larger, though producing a lower accuracy provides a more accurate and trusted estimate of the performance accuracy of the proposed approach.

Table 2: Confusion matrix for Angular view dataset using RLH feature

	Car	Jeep	Bus	Truck
Car	1480	0	0	0
Jeep	60	1380	20	0
Bus	20	60	1480	0
Truck	0	0	0	1500

Table 3: Confusion matrix for F/R view dataset using RLH feature

	Car	Bus
Car	500	10
Bus	20	470

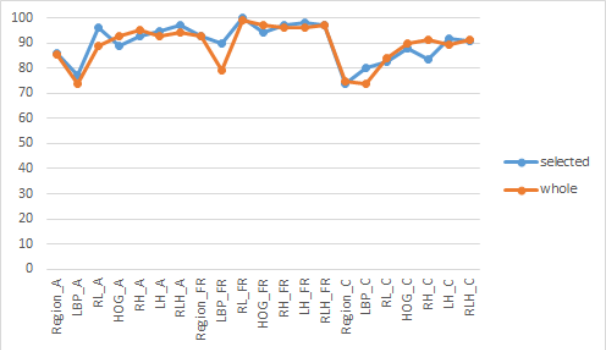

Figure 7: Accuracy on both datasets

Figure 7 plots the accuracy of various techniques, with and without feature selection, for comparison purposes. We see that the feature combination techniques, in particular the RLH technique performed generally better in all experiments.

Table 4: Speed of processing using varying feature attributes in seconds

Features	F/R - W	F/R - S	Angle-W	Angle-S	CV - W	CV - S
HOG	0.02	0.01	0.08	0.04	0.17	0.05
Region	0.01	0	0.03	0.02	0.03	0.02
LBP	0.05	0.01	0.2	0.03	0.37	0.04
RL	0.03	0.01	0.14	0.03	0.32	0.05
RH	0.01	0	0.09	0.03	0.14	0.04
LH	0.03	0.01	0.15	0.04	0.27	0.05
RLH	0.02	0.01	0.13	0.04	0.6	0.07

Using different combinations of features will result in spending different amount of time for modelling. Table 4, and Figure 8 illustrate that when the whole feature set is used, time required for modelling increased; this is due to the fact that when the number of feature attributes are large, more time is required for the modelling to complete successfully. However the careful analysis of the results also indicate that feature selection can improve the classification result and reduce the feature set to a reduced number of discriminative features that result in making the time requirement for classification minimal.

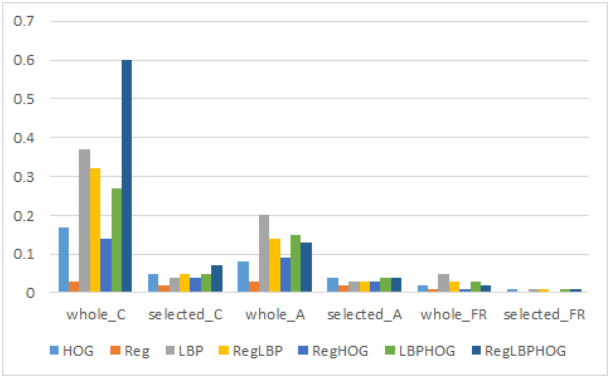

Figure 8: Speed of processing: whole vs selected. Notations used: C - combined view, FR - frontrear view, A - angular view

For the purpose of detailed analysis of the performance of the proposed approach, the classification performance is evaluated using the receiver operating characteristic (ROC) curve (see figure 9 below) that helps visualise performance, in detail. In a ROC curve the True Positive Rate (sensitivity or recall) is plotted as a function of the False Positive Rate (false alarm rate) for different cut-off points of a parameter.

True Positive Rate $= \frac{tp}{(tp+fn)}$;

False Positive Rate $= \frac{fp}{(fn+tn)}$;

where, tp denotes the number of true positives (an instance that is positive and classified as positive); tn denotes the number of true negatives (an instance that is negative and classified as negative); fp denotes the number of false positives (an instance that is negative and classified as positive) and fn denotes the number of false negatives (an instance that is positive and classified as negative).

According to [27], an ROC curve visualises the following:

1. It shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity).
2. The closer the curve follows the left-hand border and then the top border of the ROC space, the more accurate is the test.
3. The slope of the tangent line at a cutpoint gives the likelihood ratio (LR) for that value of the test.

Further the accuracy of performance is defined as:

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{16}$$

Accuracy is measured by the Area Under the ROC Curve (AUC). An area of 1 represents a perfect test; an

area of 0.5 represents a worthless test. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system [27]:

- 0.90-1 = excellent (A)
- 0.80-0.90 = good (B)
- 0.70-0.80 = fair (C)
- 0.60-0.70 = poor (D)
- 0.50-0.60 = fail (F)

In summary the ROC curve shows the ability of the classifier to rank the positive instances relative to the negative instances. The table below shows the true, false positives including the AUC values on all datasets using the RLH feature combination.

Given the above observations and facts, we plot the ROC graphs of the proposed approach when tested with the F/R datasets and angular datasets, in figures 9,10 and 11 below.

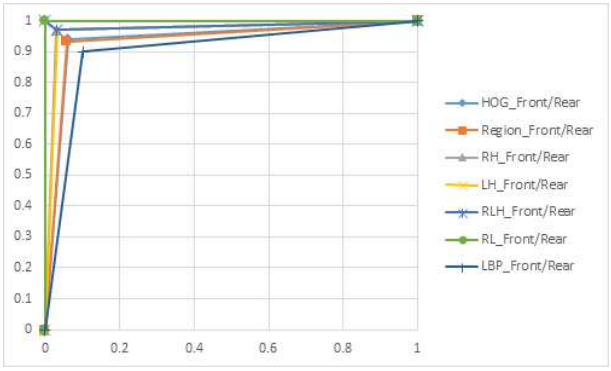

Figure 9: ROC curve of front/rear view datasets

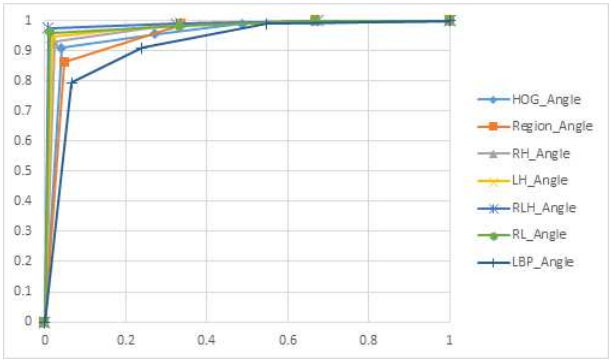

Figure 10: ROC curve of angular view datasets

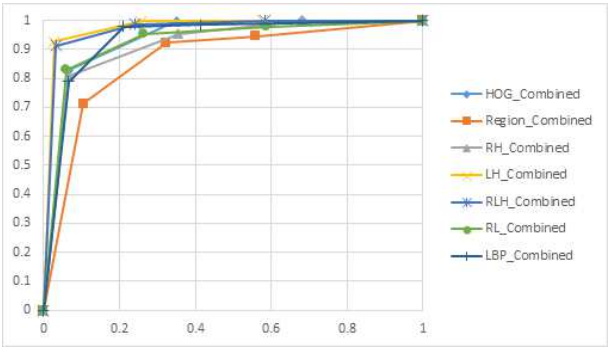

Figure 11: ROC curve of combined view datasets

The average AUC value for the classification of using the proposed feature combination on F/R and angular datasets is 97%, which is greater than 90%. Therefore the average performance (across the classification of various vehicle types) of the algorithm can be concluded to be excellent.

It is noted that each point on the ROC curve represents a TPR/FPR pair, corresponding to a particular decision threshold. The AUC is a measure of how well a parameter can distinguish between groups. ROC curves can also be used to compare performance of two or more experiments (see figures 9,10 and 11).

For the purpose of detailed analysis, we present the F-measure analysis of the classification performances in table 5. F-measure is a measure of an experiment's accuracy. It is the harmonic mean of precision and recall given as:

$$F = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{17}$$

Only the results when feature selection was used have been tabulated. From the table 5, we see the impact feature combinations on the percentage accuracy values indicated. It is clear that combining two or all three types of features enables a more accurate overall performance.

A final observation is that the accuracy levels are better when the training and testing are both done on footage captured within a limited angle. When all data is combined the accuracy drops. However this drop of accuracy is not significant to rule out that the proposed approaches will work regardless of the angle of approach of the vehicle.

Table 5: F-measure recognition percentages

Features	Angular view	Front/Rear view	Combined view	Average value
HOG	89%	94%	87.8%	90%
LBP	77%	90%	80%	82%
Region	86%	93%	74%	84%
RH	93%	97%	84%	91%
RLH	97%	97%	91%	95%
RL	96%	100%	83%	93%
LH	95%	97%	92%	94.6%

5. State of the art Comparison

In this section we present the comparison between the proposed system and the other approaches that are found in the state of the art.

An overall accuracy of 95% was recorded in classification, which could be considered as a satisfactory outcome. It is worth to noticing that two views were considered in this experiment: F/R and angular views. Comparing our results with the one reported by Ozkurt [12], we can see from Table 6 that our system returned better accuracy figures. The main drawback of this system is that experiments was performed on two views only.

The most notable contribution of the proposed technique is the classification process, which is not restricted to fixed view angle cameras. Powered by a combination of local features that are scale and rotation invariant, our system can theoretically detect vehicles and their types in video footage captured from a wide range of angles. This makes the proposed technique quite suitable to be applied in real-world scenarios where the viewing angles of traffic surveillance cameras usually change. Other techniques have been reported might, based on their own specific data, claim a higher accuracy rate than the proposed technique. However, a closer look at these techniques reveals a number of restrictions. For example, in [11], Kafai and Bhanu, the algorithm is based on features extracted from rear view images of vehicles. Moreover, specific details such as number plates and tail-lights must be visible in the image. Similar drawbacks could also be found within the data collected in Ma and Grimson [28] and in the successive experiments of Ambardekar et al [29]. Despite reporting an almost perfect result in one experiment (with the PCA-DIVS based algorithm), Ambardekar et al. admitted that their experiments were performed under a number of constraints such as, the input video must be captured from an overlooking camera; orientation, angles and road-camera distance must be pre-measured. These points clearly indicate that the technique proposed within this paper is most suitable to be directly implemented in a real-life scenario.

The comparison between different approaches and the proposed approach is presented in the following table:

6. CONCLUSIONS

In this paper we have proposed a real-time vehicle type detection and counting system that can be re-used, independent of the direction of view. The system is based of detecting a vehicle and using a combination of features of Region, Local Binary Pattern and Histogram Oriented Gradient, to identify the vehicle type. Further we show that using a suitable feature selection approach both the speed and the accuracy of the algorithms can be significantly increased. Average accuracy figures reaching 95% has been achieved on CCTV video footage captured via a general purpose, non-calibrated camera on the side of a motorway during a ten hour recording period.

We are currently working on introducing a vehicle tracking algorithm so that the vehicle type can be recognised not on a frame-by-frame basis but on a tracked object basis. This allows opportunities to further increase the robustness and the accuracy of the proposed system. More extensive testing to evaluate the performance of the algorithm under non-ideal illumination situations will also be discussed.

ACKNOWLEDGEMENTS

This work was completed with the support of Nigerian Defence Academy, Kaduna, through the Tertiary Education Trust Fund (TETFUND) intervention, Nigeria. The authors was also supported in part by the Loughborough University and the Research Council of the Sultanate of Oman, research grant agreement no [ORG SU EBR 12 013].

References

[1] Interpol, Vehicle crime (Accessed 30 April 2014).
URL `http://www.interpol.int/Crime-areas/Vehicle-crime/Vehicle-crime`

[2] S. Ramachandra T.V, Emissions from indias transport sector: Statewise synthesis, Atmospheric Environment 1 (8).

[3] Y. Li, B. Li, B. Tian, Q. Yao, Vehicle detection based on the and-or graph for congested traffic conditions, Intelligent Transportation Systems, IEEE Transactions on 14 (2) (2013) 984–993. doi:10.1109/TITS.2013.2250501.

[4] E. A. Mosabbeb, M. Sadeghi, M. Fathy, A new approach for vehicle detection in congested traffic scenes based on strong shadow segmentation, in: Advances in Visual Computing, Springer, 2007, pp. 427–436.

[5] M. Yin, H. Zhang, H. Meng, X. Wang, An hmm-based algorithm for vehicle detection in congested traffic situations, in: Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE, IEEE, 2007, pp. 736–741.

[6] S. Gupte, O. Masoud, R. Martin, N. Papanikolopoulos, Detection and classification of vehicles, Intelligent Transportation Systems, IEEE Transactions on 3 (1) (2002) 37–47. doi:10.1109/6979.994794.

[7] P. Daigavane, P. Bajaj, Real time vehicle detection and counting method for unsupervised traffic video on highways, International Journal of Computer Science and Network Security 10 (8).

[8] C.-C. R. Wang, J.-J. Lien, Automatic vehicle detection using local features;a statistical approach, Intelligent Transportation Systems, IEEE Transactions on 9 (1) (2008) 83–96. doi:10.1109/TITS.2007.908572.

[9] H. Ranga, M. Ravi Kiran, S. Raja Shekar, S. Naveen Kumar, Vehicle detection and classification based on morphological technique, in: Signal and Image Processing (ICSIP), 2010 International Conference on, 2010, pp. 45–48. doi:10.1109/ICSIP.2010.5697439.

[10] J. Y. Ng, Y. H. Tay, Image-based vehicle classification system, arXiv preprint arXiv:1204.2114.

[11] M. Kafai, B. Bhanu, Dynamic bayesian networks for vehicle classification in video, Industrial Informatics, IEEE Transactions on 8 (1) (2012) 100–109.

[12] C. Ozkurt, F. Camci, Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks, Mathematical and Computational Applications 14 (3) (2009) 187.

[13] D. Reynolds, Gaussian mixture models, Encyclopedia of Biometrics (2009) 659–663.

[14] D. OpenCV, Background subtraction (Accessed 23rd January 2014).
URL http://docs.opencv.org/trunk/doc/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html
[15] J. Parker, Algorithms for Image Processing and Computer Vision, 2nd Edition, Wiley Publishing, inc, Indianapolis, 2011.
[16] R. C. Gonzalez, R. E. Woods, Digital image processing, 3rd Edition, Pearson Prentice Hall, Dorling Kindersley, 2008.
[17] D. Huang, C. Shan, M. Ardabilian, Y. Wang, L. Chen, Local binary patterns and its application to facial image analysis: a survey, Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 41 (6) (2011) 765–781.
[18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 886–893.
[19] M. A. Hall, L. A. Smith, Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper, in: FLAIRS Conference, 1998, pp. 235–239.
[20] Y. Lu, K. Boukharouba, J. Boonært, A. Fleury, S. Lecœuche, Application of an incremental svm algorithm for on-line human recognition from video surveillance using texture and color features, Neurocomputing 126 (2014) 132–140.
[21] C. Nakajima, M. Pontil, B. Heisele, T. Poggio, Full-body person recognition system, Pattern recognition 36 (9) (2003) 1997–2006.
[22] S. Milan, H. Vaclav, B. Roger, Image Processing Analysis, and Machine Vision, 3rd Edition, Cengage Learning, Delhi, 2008.
[23] F. Lauer, Y. Guermeur, MSVMpack: a multi-class support vector machine package, Journal of Machine Learning Research 12 (2011) 2269–2272, http://www.loria.fr/~lauer/MSVMpack.
[24] R. C. Gonzalez, R. E. Woods, S. L. Eddins, Digital image processing using MATLAB, 2nd Edition, Vol. 2 of Digital image processing using MATLAB, second edition, Tata McGraw Hill, Haryana, 2009.
[25] C. Fosu, G. Hein, B. Eissfeller, Determination of centroid of ccd star images, Int. Arch. Photogram. Remote Sens. Spatial Inform. Sci 35 (2004) 612–617.
[26] N. C. Infrastructure, Image processing toolbox (Accessed 9 June 2014).
URL https://nf.nci.org.au/facilities/software/Matlab/toolbox/images/regionprops.html
[27] M. Thomas G. Tape, Interpreting diagnostic tests (retrieved 15th June, 2014).
URL http://gim.unmc.edu/dxtests/roc3.htm
[28] X. Ma, W. E. L. Grimson, Edge-based rich representation for vehicle classification, in: Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on, Vol. 2, IEEE, 2005, pp. 1185–1192.
[29] A. Ambardekar, M. Nicolescu, G. Bebis, Efficient vehicle tracking and classification for an automated traffic surveillance system, in: Proceedings of the 10th IASTED International Conference, Vol. 623, 2008, p. 101.
[30] C. C. R. Wang, J. J. J. Lien, Automatic vehicle detection using local features x2014;a statistical approach, IEEE Transactions on Intelligent Transportation Systems 9 (1) (2008) 83–96. doi:10.1109/TITS.2007.908572.

Table 6: Comparison between approaches

Techniques	Limitations
Method in [30]	Only detection, no classification. Fixed view (F/R). Roof, headlights, tail-lights details must be visible.
Method in [12]	Fixed view (Top). Manually cropped video frames.
Method in [11]	Fixed view (Rear). LP, tail-lights details must be visible.
Method in [3]	Only detection, no classification. Fixed view (F/R). Many assumptions and time consuming.
Method in [28]	Fixed view (overlooking); Angle and distant must be measured.
Method in [29]	Orientation, angles and road-camera distance must be pre-measured
Our approach	Experiments performed on two views only